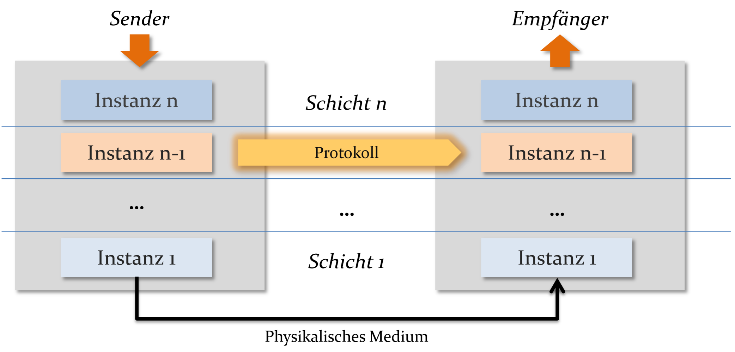
**Strukturierung**



**Strukturierung:**

* Einteilung von Aufgaben in Schichten
  + Jede Schicht bietet darüberliegender Dienste an
  + Instanzen einer Schicht interagieren über Protokolle
  + Schichten von oben nach unten:

7. Anwendung *anwendungsspezifische Protokolle*

6. Darstellung *betriebssystemunabhängiges Darstellungsformat*

5. Sitzung *Verwaltung von Sitzungen*

4. Transport *Ende-zu-Ende Transportdienst (TCP, UDP…)*

3. Vermittlung *Adressierung und Routing von Paketen*

2. Sicherung *Verpackung in Datenrahmen, versenden der Datenrahmen*

1. Bitübertragung *Datenübertragung über Medium*

**Grundlagen der Datenübertragung**

**Einführung:**

* Bitübertragungsschicht: **physical layer**
  + physikalische Übertragung von Informationen zwischen Sender & Empfänger
  + Übertragungsmedium und Energieform
    - Daten werden in geeignete Energieform gewandelt, über Medium übertragen, zurückgewandelt
* Transceiver (Transmitter + Receiver) zur Daten-/Energieübertragung 🡪 wandelt Daten in Energie um
* Medium: Übertragungswerkzeug für verschiedene Energieformen (Kupferkabel, Luft, …)

**Duplex:**

* Simplex-Verbindung:
  + Daten von Sender zu Empfänger, kein Rückkanal
  + z.B. Radio, Vorlesung
* Halbduplex-Verbindung:
  + Abwechselndes Senden / Empfangen, Datenrichtung ändert sich
  + z.B. Walkie-Talkie
* Vollduplex-Verbindung:
  + Gleichzeitig Senden und Empfangen, Medium bidirektional benutzt
  + z.B. heutiges Ethernet, Telefon

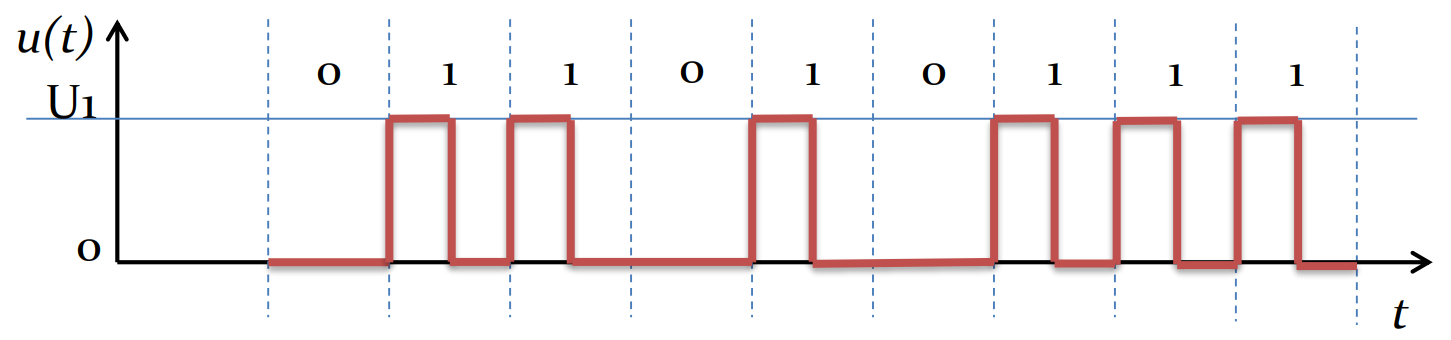
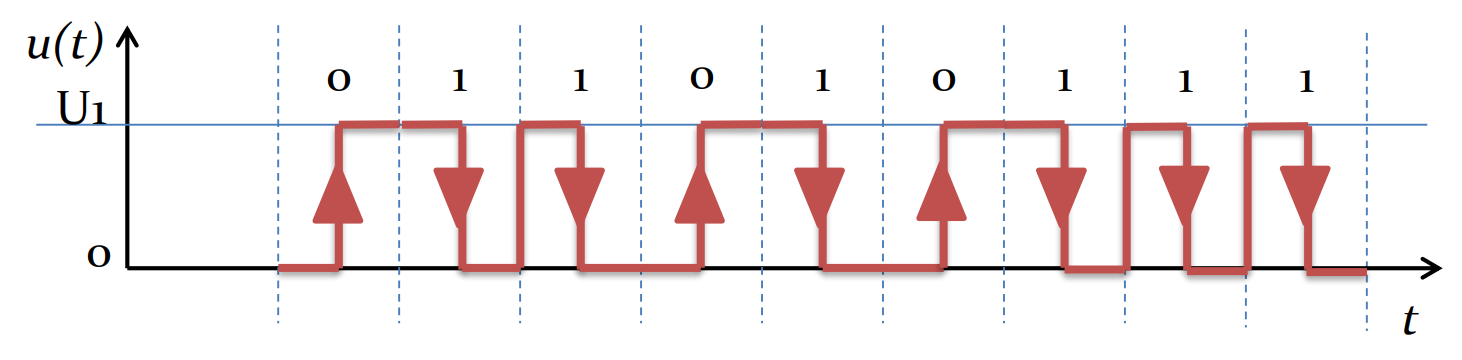
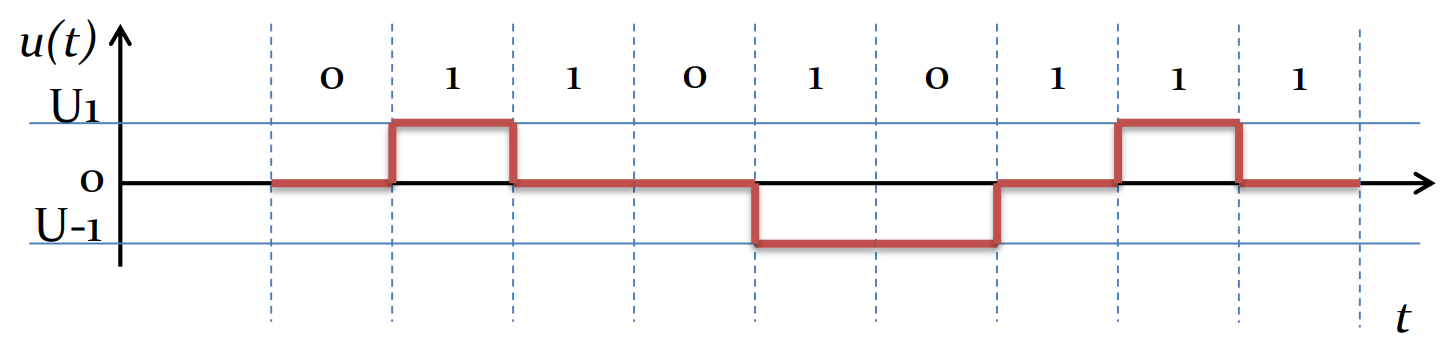
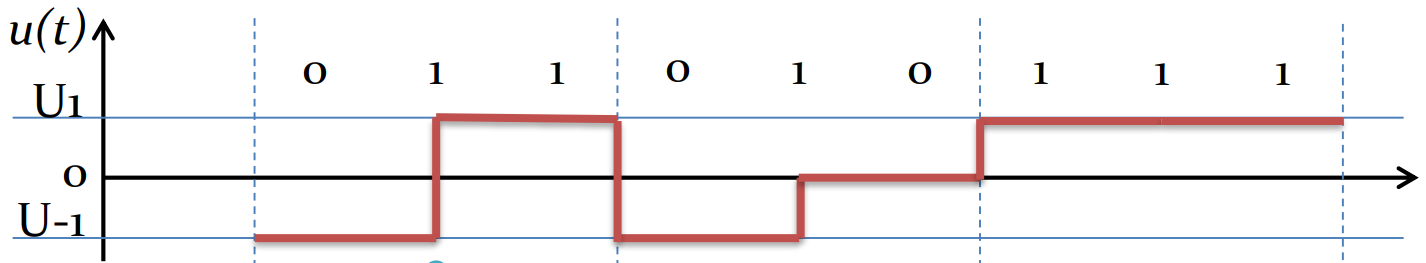
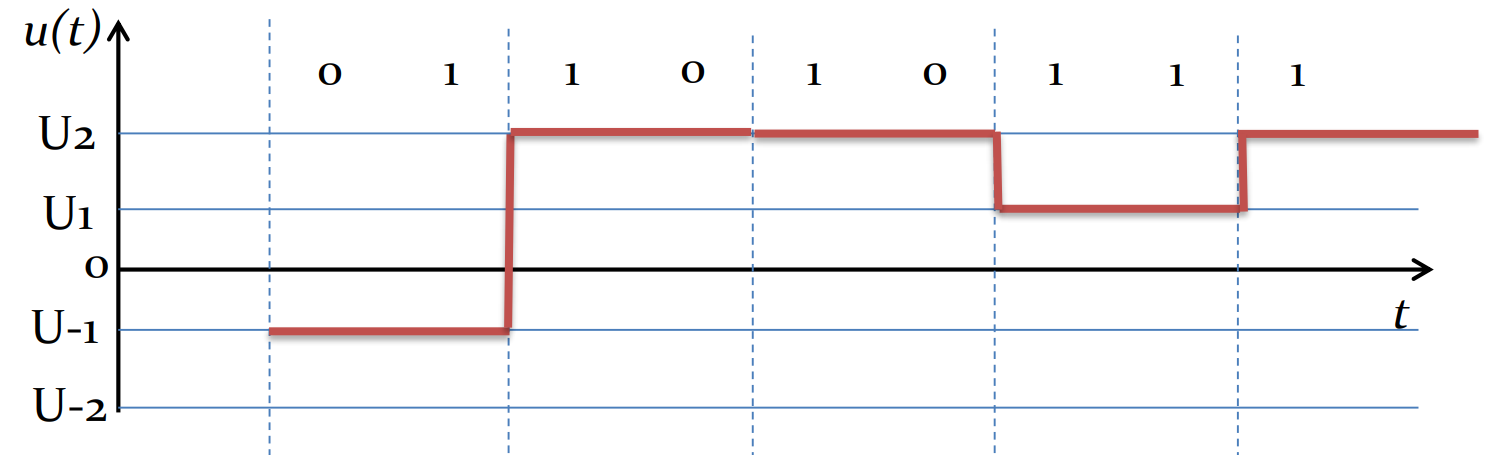
**Datenrate:**

* R = Bitrate, N = Anzahl gesendeter Bits, t = benötigte Sendezeit
* **Laufzeit**: Verzögerungszeit zwischen Versenden und Ankunft der Daten (Latenz)

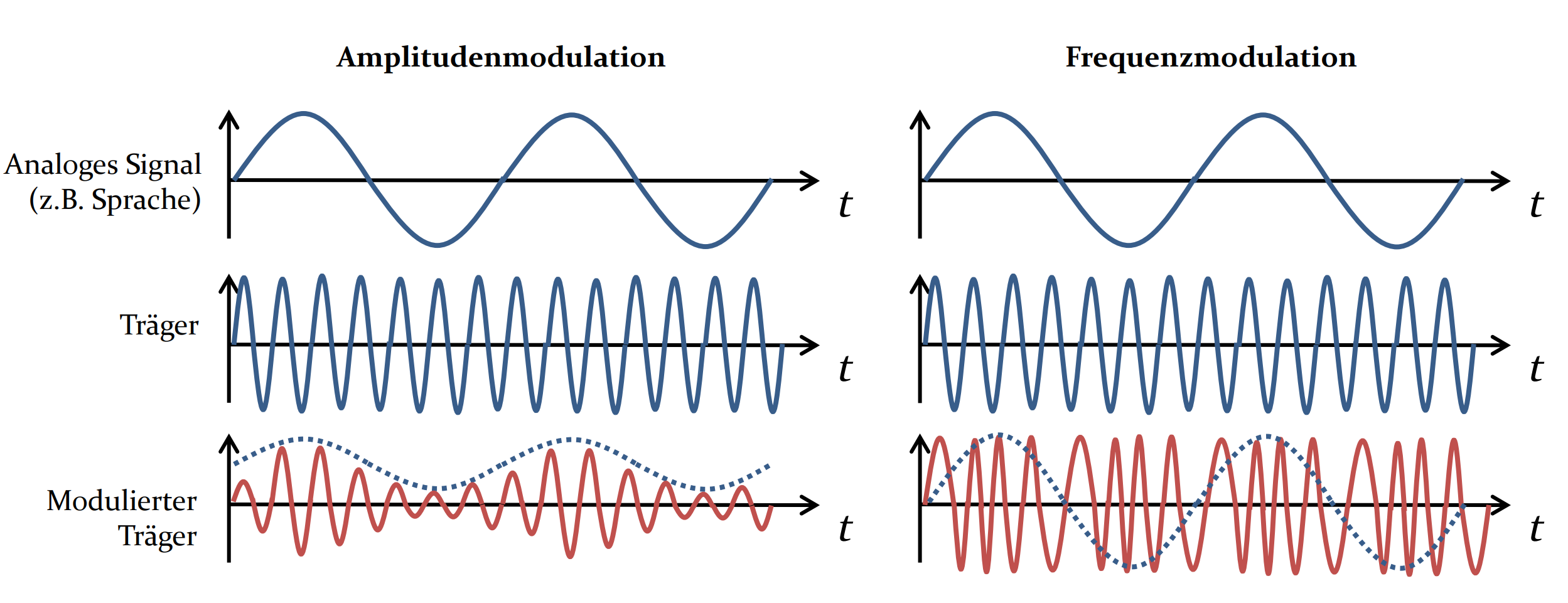
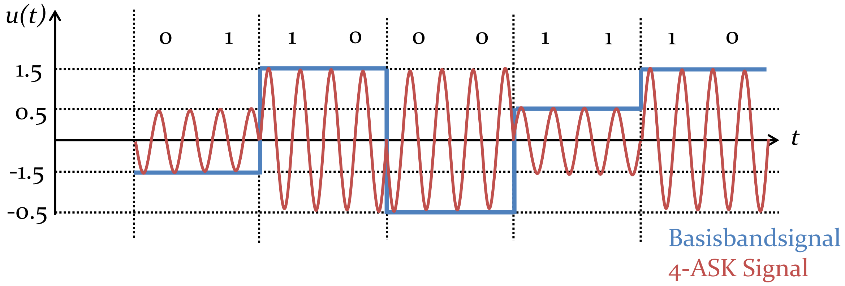
**Bitfehlerrate:**

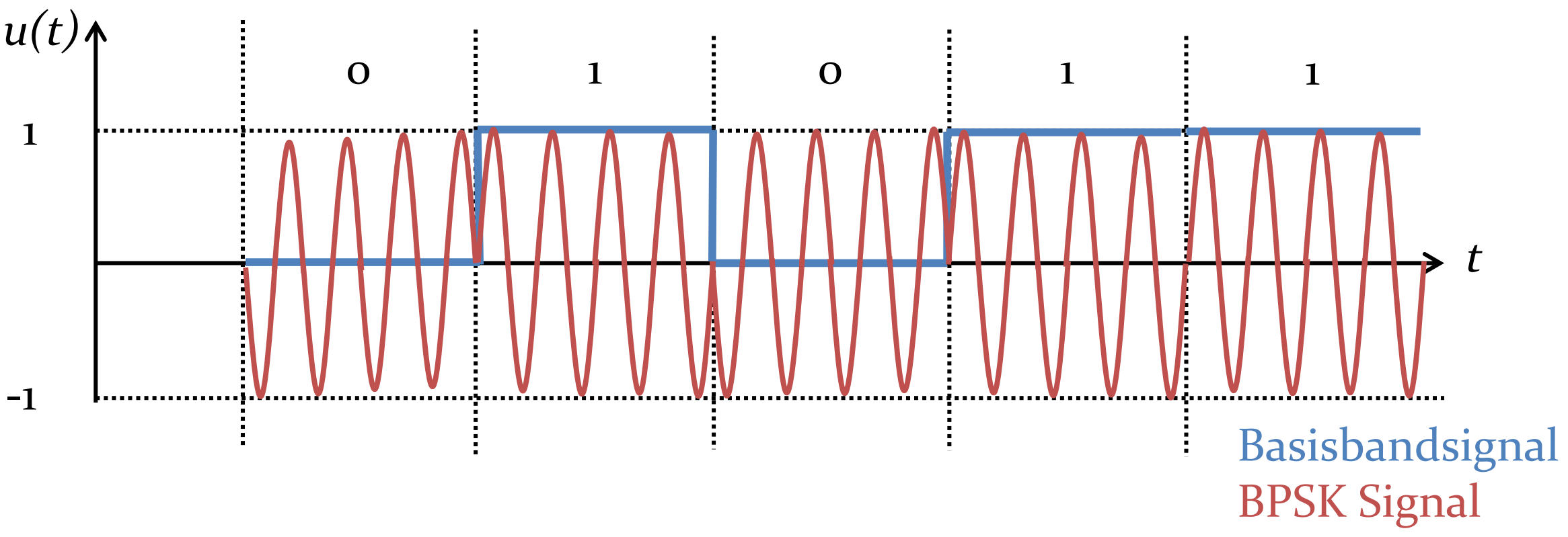
* BER = Bit Error Rate, NERR = Anzahl fehlender Bites, NTOTAL = Anzahl gesendet Bits
* Gründe: Störeinflüsse, Dämpfung durch langen Kanal …
* Verbesserung: verschiedene Kanalkodierungsverfahren, Verbesserung Übertragungstechnik

**Kodierungsarten:**

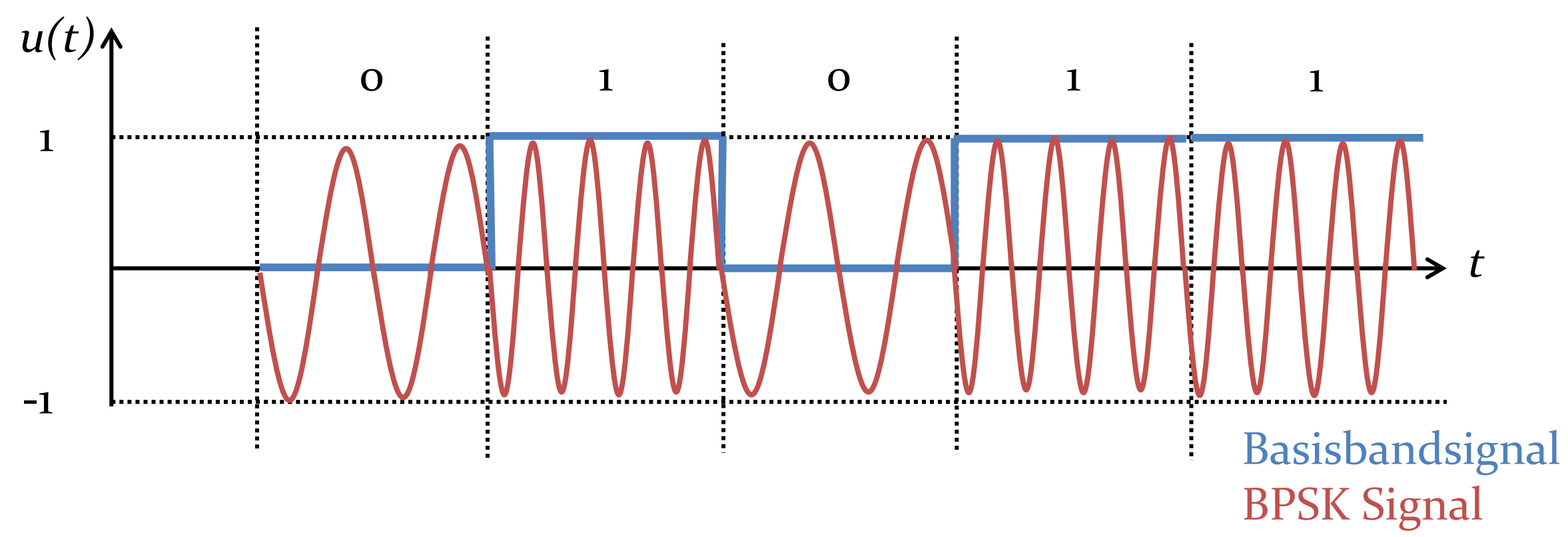
* Kanalkodierung: Hinzufügen von Redundanz, Datenverlust wird vermindert
  + Erhöhung Hamming-Distanz: MindestAnzahl Bits, um die sich 2 Codewörter unterscheiden
    - z.B.: A = 001, B = 010, Hamming-Distanz = 2 (2 Bits um A zu B zu machen)
    - Bitfehler erkennen und Bitfehler korrigieren
  + Coderate: k = Anz. Symbole einer Nachricht, n = Länge des Codewortes
    - Geringe Coderate: geringe Datenrate, besser Fehlerkorrekturfähigkeit
    - Vorwärtsfehlerkorrektur: **FEC**, Kanalcode kann Fehler erkennen und beheben
    - Rückwärtsfehlerkorrektur: **ARQ**, Fehler kann nur erkannt werden, Korrektur: neue Übertragung
* Leitungskodierung Datenrate R, Symbolrate S, Bandbreite B
  + Leitungscodes möglichst hohe Datenraten in möglichst wenig Spektrum 🡪 wegen Grenzfrequenz
  + Fourier-Transformation Umwandlung zwischen Zeit- und Frequenzbereich
  + Signalspektrum:
    - Bandbreite: gibt den Frequenzbereich des Signalspektrums an
  + Arten von Leitungskodierungen:
    - NRZ (Non-Return-To-Zero): S = R = B
      * nicht gleichanteilsfrei
      * schlechte Taktrückgewinnung
      * 1 = volles Feld, 0 = leeres Feld
    - RZ (Return-To-Zero): S = 2\*R, B = 2\*R
      * einfachere Takrückgewinnung
      * doppelte Signalbandbreite zu NRZ
      * Problem langer Nullfolgen
      * 1 = halbvolles Feld, 0 = leer
    - Manchester / BiPhase: S = 2\*R, B = 2\*R
      * Kein Problem mit langen Nullfolgen
      * gut Taktrückgewinnung
      * 1 = linke Hälfte, 0 = rechte Hälfte
    - MLT-3 (Multi-level Transmission Encoding) S = R, B = ¼ R
      * Niedrige Signalbandbreite
      * Gleichanteilsfrei
      * Nullfolgen-Problem
      * Änderung bei 1, keine Änderung bei 0
    - Ternäre Codes S = 2/3 R
      * Senkung der Bandbreite
      * Umwandlung Binär in Tenär (z.B. 0 1 1 = - + )
    - Quaternäre Codes S = ½ R
      * Weitere Reduktion der Bandbreite
      * Umwandlung Binär in Quaternär (z.B. 0 1 = -1)
      * Verwendung z.B. bei Gigabit Ethernet oder ISDN

**Modulation:**

* Basisbandsignal durch Modulation in höheren Frequenzbereich verschieben
* ermöglicht z.B. Übertragung von Funk (Medium ist geteilter Kanal)
  + mehrere Übertragungssysteme teilen sich verfügbares Spektrum 🡪 Frequenz-Multiplex / FDMA
  + Signal wird mit Trägersignal multipliziert 🡪 wird in Frequenzbereich des Trägers verschoben
* Analoge Modulation (Rundfunk, analoger Sprechfunk)
  + **Amplitudenmodulation** (AM)
  + **Frequenzmodulation** (FM)
* Digitale Modulation
  + kodierte Impulse eines Digitalsignals werden zur Modulation verwendet (siehe Leitungskodierung)
  + **Amplitude Shift Keying (ASK):** 
    - Amplitude wird an Höhe des Basisbandsignals angepasst



* + **Phase Shift Keying (PSK):** 
    - Phase wird verändert 🡪 bei 0 startet Frequenz „nach unten“, bei 1 „nach oben“



* + **Frequence Shitf Keying (FSK):**
    - Frequenz des Trägersignals wird durch Basisbandsignal umgetastet
* **Quadraturamplituendmodulation** **(QAM)**
  + statt einem werden zwei orthogonale Träger (Sinus und Cosinus) amplitudenmoduliert und anschließend addiert 🡪 wie wenn Amplitude und Phase gleichzeitig moduliert werden

**Medien**

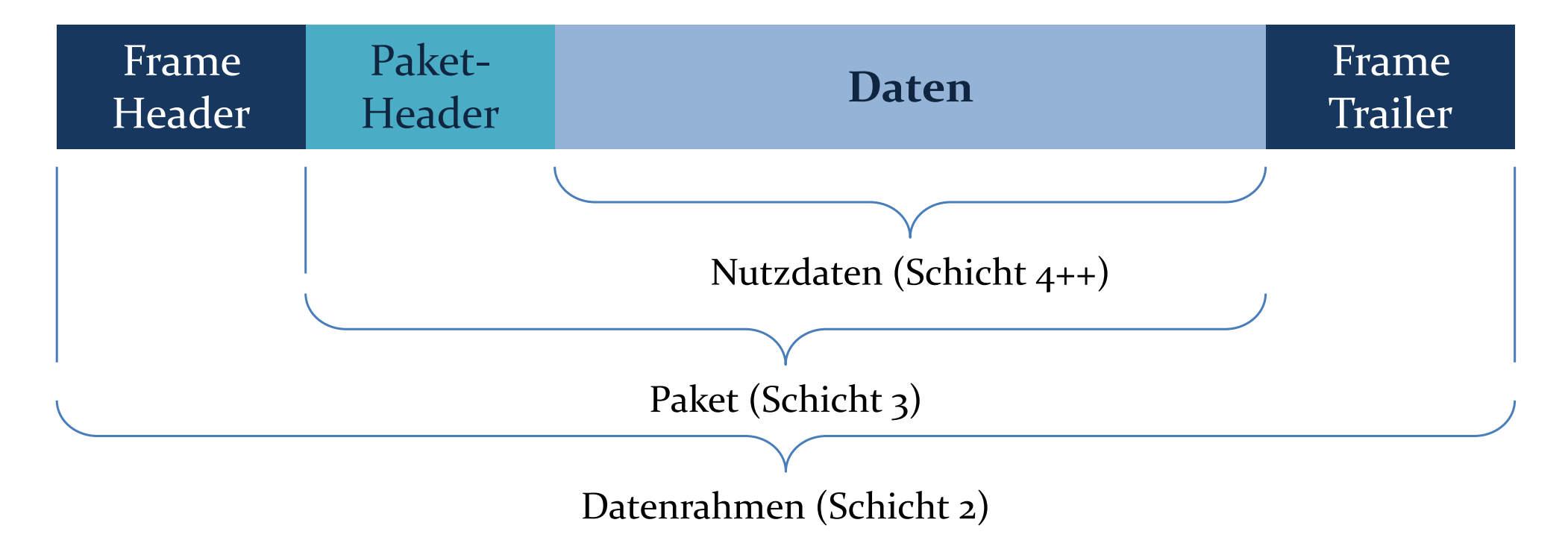
* **Twisted** **Pair**: Verdrilltes Kupferkabel
  + CAT 5e: Netzwerkkabel aus Heim-/Bürovernetzung
  + Elektrische Signale
* **Verdrilltes Adernpaar**
  + Differentiale Signale
  + Nutzsignal ist Spannungsdifferenz beider Drähte eines Pärchens
* **Koaxialkabel**
  + Klassisches Antennen- / Fernsehkabel
  + Wellenleiter, Signalasubreitung in Form einer TEM-Welle
* **Lichtwellenleiter**
  + Ausbreitung eines Lichtbündels entlang der Glasfaser
  + für WAMs (Wide Area Networks)

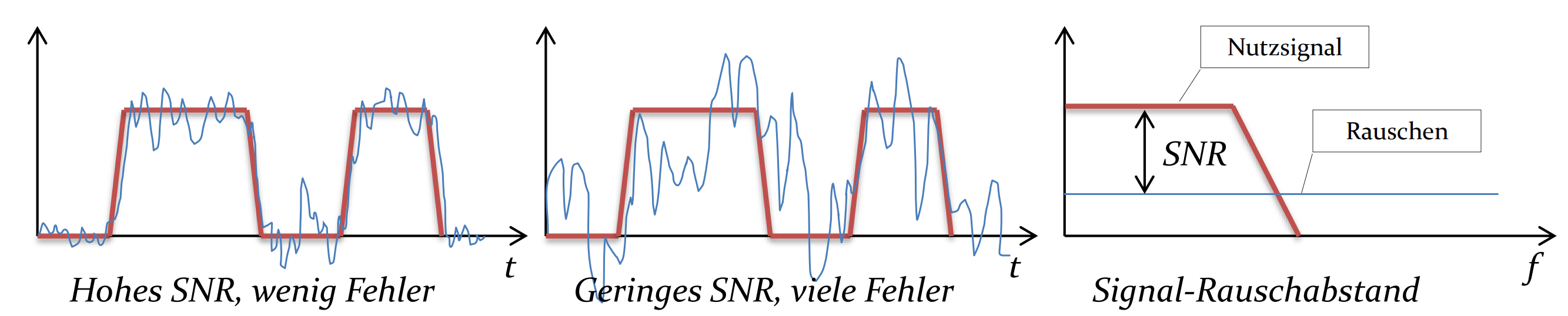
**Ethernet Physical Layer Basics**

* 100BASE-TX Fast Ethernet:
  + verhält sich wie Full Duplex, sind aber 2 Simplex Kanäle
* 100BASE-T Gigabit Ethernet
  + Vier Vollduplex-Kanäle
  + Daten werden über alle vier Kanäle in beide Richtungen gesendet

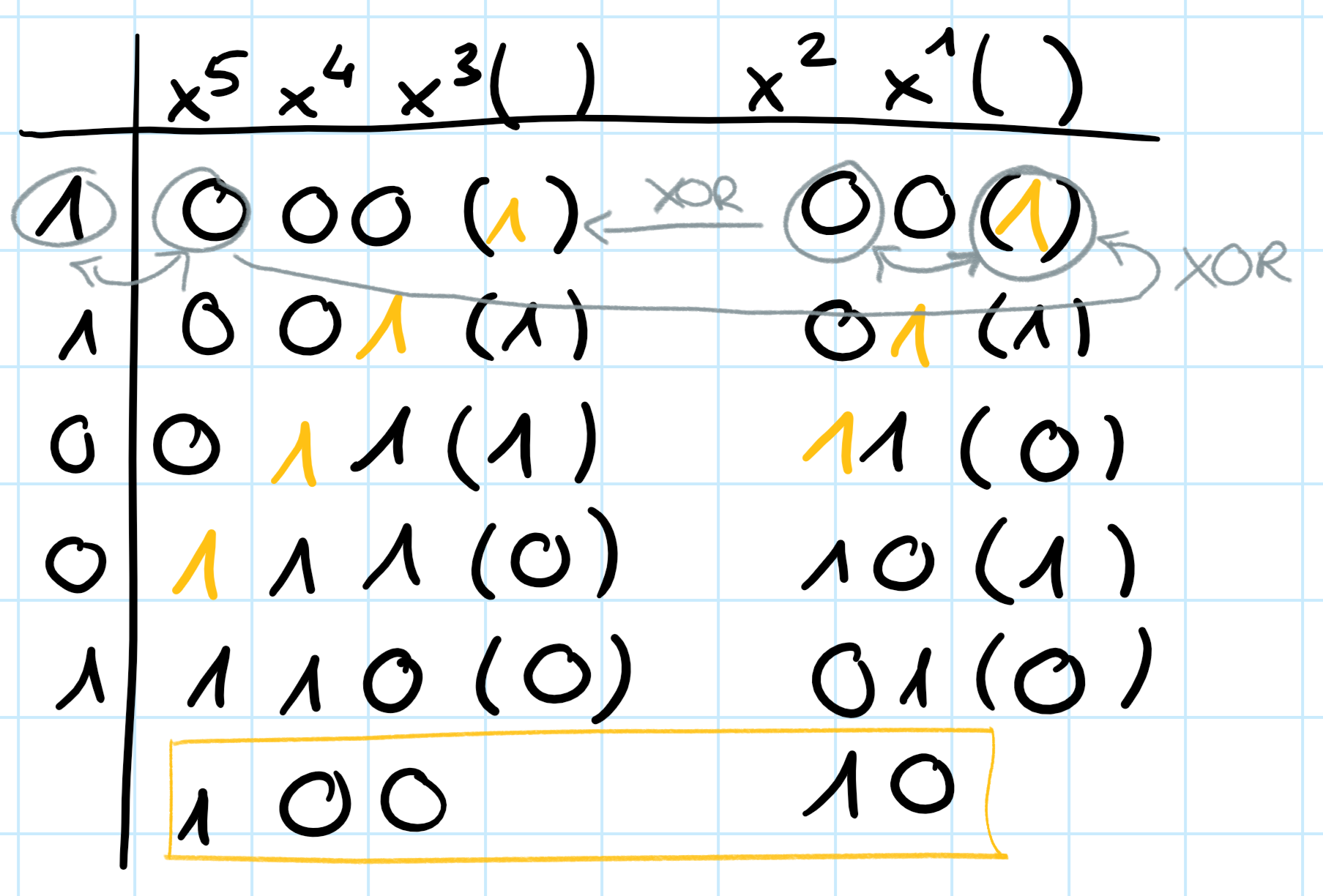
**Sicherungsschicht**

**Pakete und Frames**

* Erklärung:
  + Daten werden in kleine Einheiten (Pakete) aufgeteilt und stückweise übertragen
  + Time Dvision Multiple Access (TDMA) (auch z.B. Scheduler bei Prozessorzugriff)
* Vorteile:
  + fairer Zugriff aller Teilnehmer gemeinsam genutzte Ressource Netzwerker
  + Responsivität des Systems
  + Fehlerhafte Datenübertragung: Empfänger prüft jedes Paket, im Fehlerfall 🡪 1 Paket neu übertragen
  + Effiziente Netzwerknutzung: Prozess sendet nicht dauerhaft, Leerzeiten nutzbar
    - Hohe Auslastung des Netzwerks möglich
* Definition:
  + Frame: Übertragungseinheiten der Sicherungsschicht (2)
    - MTU (maximale Nutzdatenlänge) : 1500 Byte
    - maximale Framelänge: 1518 Byte
    - minimale Framelänge: 64 Byte (CSMA)
  + Paket: Übertragungseinheit der Vermittlungsschicht (3)
  + Daten in Paketen aufgeteilt, in Frames übertragen
  + Header für Metadaten, Adressierung, Anwendungszuordnung…
  + Netzwerkpaket beinhaltet mehrere Prokolle / Schichten
* OSI-Model: Schicht n 🡪 nur eigene Headerdaten relevant
  + Encapsulation: beim Versenden, Header an Schicht anfügen, an darunterliegende weitergeben
  + Decapsulation: beim Empfangen, Header jeder Schicht entfernen, an darüberliegende weitergeben
* MAC-Adresse: weltweit eindeutig
  + 6 Byte lang, von IEEE vergeben, erste 3 Byte OUI (Organization Unique Identifier)
  + spezielle Adressen: I/G-Bit (Individual/Group) gibt an ob Uni- oder Multicast-Adresse
    - Unicast: Eindeutige Adresse einer Station (normale MAC-Adresse)
    - Multicast: zuvor festgelegte Gruppe an Stationen
    - Broadcast: alle Stationen dieses LANs/WLANs werden adressiert



**Fehlererkennung**

* Übertragungsfehler:
  + Elektromagnetische Störungen 🡪 Rauschen = Hintergrundstörungen (SNR = Signalrauschabstand)
* Mechanismen Fehlererkennung:
  + **Paritätsbit** Festlegung auf gerade/ungerade Anzahl ‚1‘-Bits
    - Prüfung: 2-dimensionale Paritätsprüfung 🡪 ein Bitfehler korrigierbar (gleichzeitig Zeilen-/Spaltenprüfung)
    - Vorteile: nur ein Bit zusätzlich
    - Nachteile: gerade Anzahl Bitfehler werden NICHT erkannt
    - Anwendung: serielle Schnittstellen
  + **Checksum** Prüfsumme üer alle Bytes
    - Prüfung: Daten werden als Ganzzahlen interpretiert und summiert
    - Vorteile: wenig Overhead, einfache Berechnung
    - Nachteile: nicht alle Fehler können gefunden werden
    - Anwenung: Netzwerkprotokolle, z.B. IP
  + **CRC** Polynomdivision durch Prüfpolynom (Zuverlässiger als Checksum) (**C**yclic **R**edundancy **C**heck)
    - Prüfung: z.B. CRC-4: 4 0er Bits anhängen, durch 10011 teilen, Rest (4-Bit lang) an Daten anhängen
    - Vorteil: Einfache Realisierung in Hardware durch XOR
    - Anwendung: Ethernet, USB, ISDN,…

**Fehlerbehandlung**

* FEC: Redundanz in jedem Paket, Prüfzahl immer mitübertragen
* Retransmission: Bei Fehlererkennung 🡪 erneute Übertragung
  + Automatic Repeat Request (ARQ): beschädigtes oder verlorenes Paket muss neu verschickt werden (Signalisierung durch Quittung (ACK) des Empfängers)
    - **stop-and-wait ARQ** erneute Sendung erst bei empfangener Quittung oder RTO (Timeout)
      * Auslastung:
        + bei hoher Laufzeit und/oder kurzen Frames sehr ineffizient
      * Spezialfall von Sliding Window, Sende & Empfangsfenstergröße 1
    - **Sliding Window** n Frames senden ohne auf ACK zu warten (Sendefenster), mit jedem ACK um ein Frame weiterverschieben, hohe Auslastung = hohe Effizienz
      * **Go-back-n ARQ** Sliding Window mit Sendefenster > 1 und Empfangsfenster = 1
        + Empfänger akzeptiert nur das nächste Paket, bei Fehler/Verlust muss ab jeweiligem Paket alles neu gesendet werden
        + z.B. TCP
      * **Selctive Repeat ARQ** Allgemeinster Fall von SW: Sendefenster = Empfangsfenster
        + Empfänger teilt Sender die Sequenznummer ungültiger Pakete zu 🡪 entsprechendes Paket neu senden
        + z.B. TCP SACK (Selective ACK) = Selective Repeat ARQ

**Technologien Lokaler Netzwerke (LANs)**

**Definition / Einteilung**

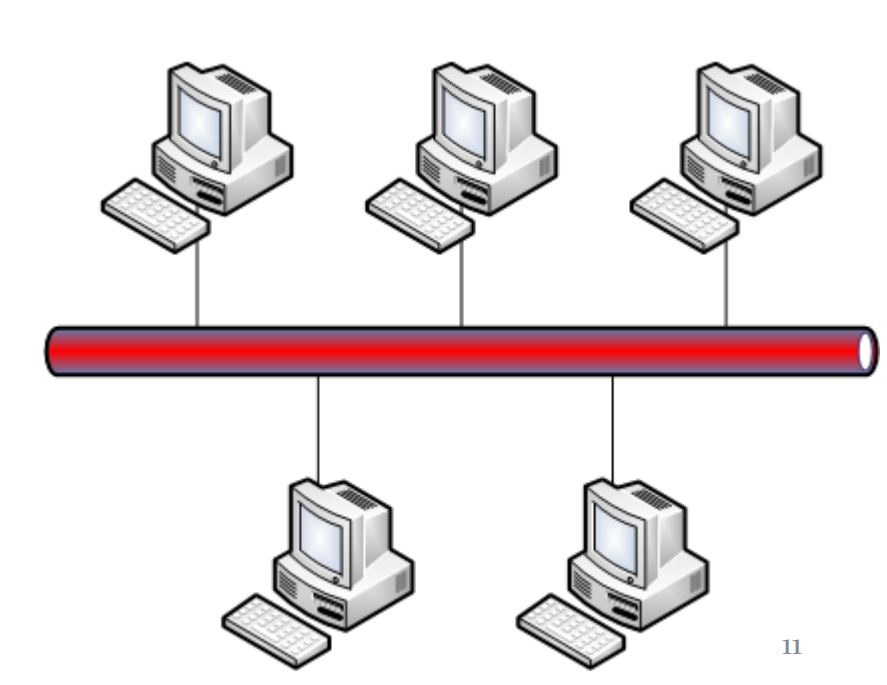
* LAN = Local Area Network

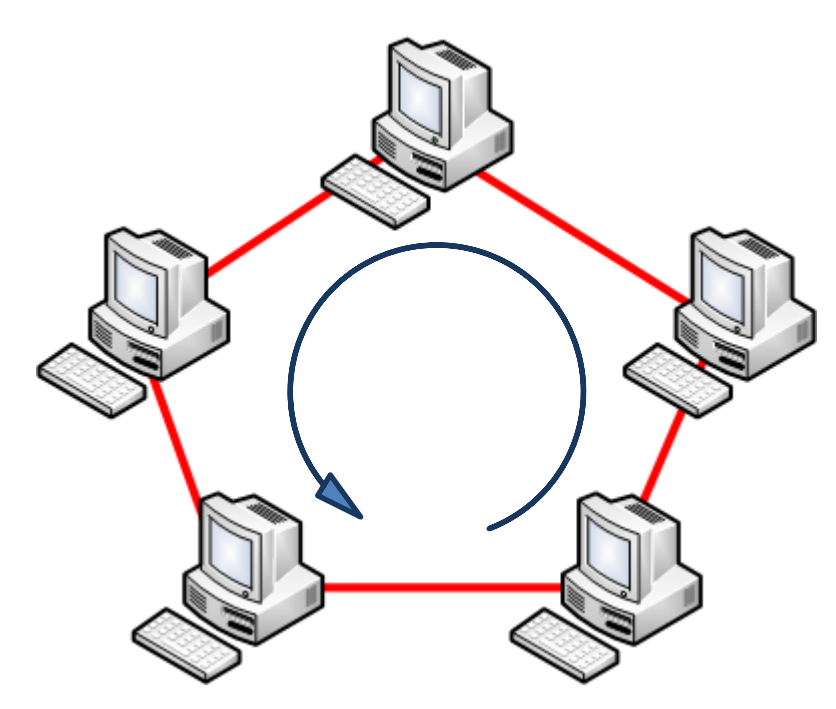
**Aufgaben des Netzwerks**

* Adressierung (wohin muss eine Nachricht)
* Kollisionsvermeidung (Zugriffskontrolle auf gemeinsame Ressourcen)
* Topologie (hohe Kapazität, Effizienz, Bezahlbarkeit) 🡪 Struktur
* Balance zwischen Skalierbarkeit, Kosten und Geschwindigkeit

**Punkt-zu-Punkt Topologie**

* Vollvermaschtes Netz (Verbindung von allen Knoten untereinander)
  + n Stationen, k Verbindungn
* für Fernnetze (WANs)

**Topologien mit geteilten Ressourcen**

* Bus:
  + alle Teilnehmer an ein gemeinsames Kabel
  + Einfache Kabelführung, wenige Leitungen
  + Kollisionen, keine 100% Auslastung möglich
  + z.B. CAN, Half-Duplex Ethernet
* Ring
  + jeder Teilnehmer mit 2 anderen verbunden (Ringform)
  + keine Kollisionen
  + niedrige verfügbare Kapazität
  + IBM Token Ring (Logisch), FDDI
* Stern
  + jeder Teilnehmer mit Switch verbunden
  + keine Kollisionen, volle Kapazität für jeden Knoten
  + Single Point Failure
  + z.B. Modernes Full Duplex Ethernet

**Half-Duplex Ethernet**

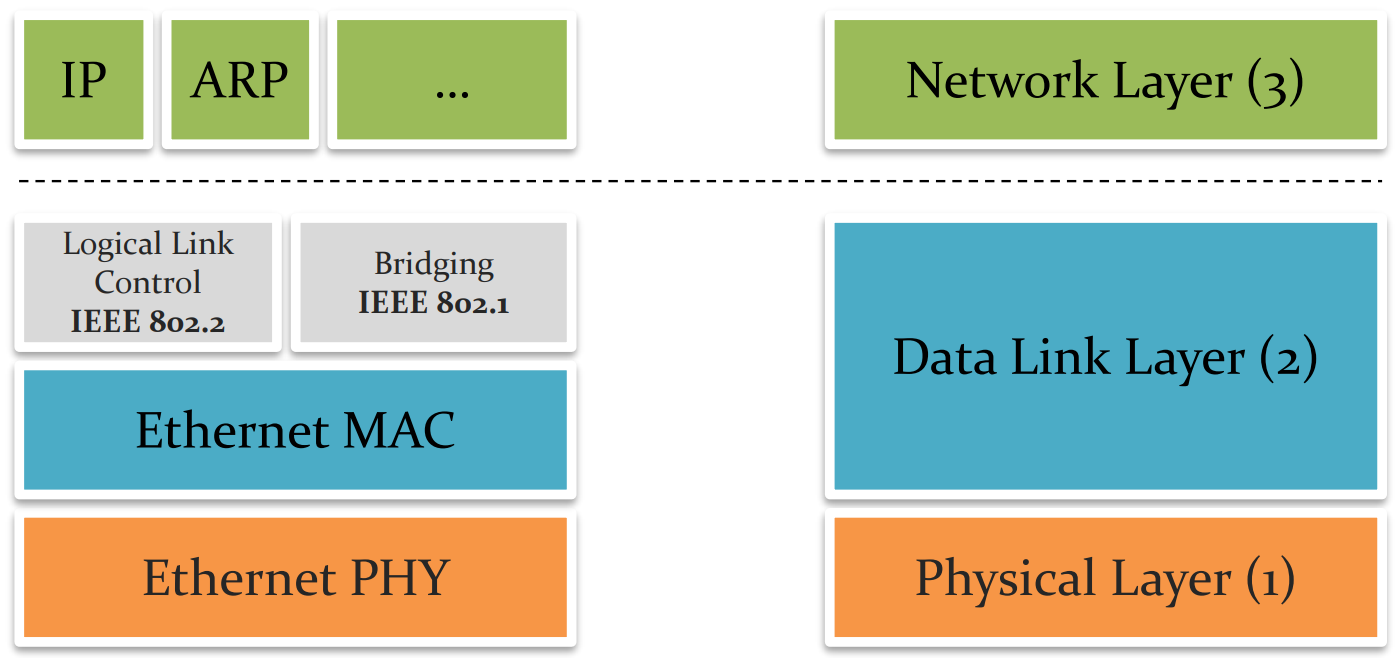
* Bus-Topologie
* Während Quellrechner sendet, müssen alle anderen „ruhig“ sein
* Kollisionserkennung:
  + CSMA/CD
    - Medium prüfen ob gerade gesendet wird
    - Sender prüft während Übertragung ob die sendenden Bits seinen entsprechen 🡪 wenn nicht = Kollision
    - zufällig lange Zeit warten & Neustart

**Der Ethernet-Frame**

* Präambel: wiederholte 010101 zur Uhr-Sync.
* SFD: Startzeichen
* MAC-Adressen: 6 Byte Ziel/Quelladresse
* Length: Länge des Pakets (>1536: Ethertype)
* Padding: Füllzeichen, Mindestlänge 64 Byte
* FCS: CRC-32 Checksumme

**Full-Duplex Ethernet**

* Stern- statt Busstruktur
* CSMA/CD ist noch integriert, wird wegen Switches aber nicht mehr benötigt
* Switch leitet Paket anhand Zieladresse weiter
  + mit Puffer und Warteschlangen
  + Store and Forward Frame wird komplett empfangen, gespeichert und wieder versendet
  + Cut Through Switch beginnt Frame auf Zielport zu versenden, sobald er Ziel MAC-Adresse hat
* Switch lernt MAC-Adresse von Ports beim einschalten
  + wenn DST-Adresse noch unbekannt ist 🡪 Frame wird an alle Ports gesendet (Flooding)
* Virtual LANs
  + Seperation des LANs im Switch durch virtual LANs (VLAN), Stationen im selben VLAN können miteinander kommunizieren, Broadcast nur für jeweiliges VLAN
  + Port-based VLAN
    - Ports sind VLAN zugewießen, Enknoten wissen nichts über VLAN
    - jedes VLAN muss bei Verbindung von Switches extra verbunden werden
  + Tagged VLAN
    - VLAN-Zuordnung in zusätzlichem Header-Feld
    - für Verbindung von Switches besser geeignet
  + Layer-3 Switches können zwischen VLANs routen
  + Vorteile:
    - hohe Flexibilität, Partitionierung in Subnetze nicht von räumlicher Lage abhängig, schnell änderbar, Isolation sensibler Daten



IEEE 802.3

**WAN-Technologien**

* ATM (Asynchronus Transfer Mode)
  + WAN, LAN
  + Switched, Multiplexverfahren, für DSL
* SONET/SDH (Synchronus Optical Network, Synchronus Digital Hierarchy)
  + WAN
  + Multiplexverfahren für optische digitale Datenübertragung
  + Backbone-Infrastruktur vieler Provide, ATM wird darüber transportiert

**Internetworking**

**Einführung**

* Zusammenschluss unzähliger LANs und WANs, mit der Aufgabe des weltweiten Datenaustausches

**Internetworking**

* zwei unabhängige Netze werden mithilfe eines Routers verbunden 🡪 Schicht 3 = Vermittlungsschicht
  + Router besitzt verschiedene Netzwerkinterfaces, für jedes Netz eine
  + pro Netz Sicherungs- und Bitübertragungsschichtinstanz
  + Wegewahl anhand weltweit einheitlicher, hierarchischer Netzwerkadressen = Routing
* schnell komplexe, vermaschte Struktur aus durch Router verbundenen Netzten
  + Router sind Knoten, Netze sind Kanten
  + Wegewahl: Router pflegt seine Routingtabelle, „**nextHop**“ anhand verschiedener Kriterien
    - Verfahren zur Bestimmung heißt **Routing-Algorithmus**
  + **Routing-Protokolle** für Austausch von Routing-Informationen zwischen den Routern

**Die Struktur des heutigen Internetzes**

Hierarchischer Aufbau:

Lokale Netze und Endsysteme (LAN)

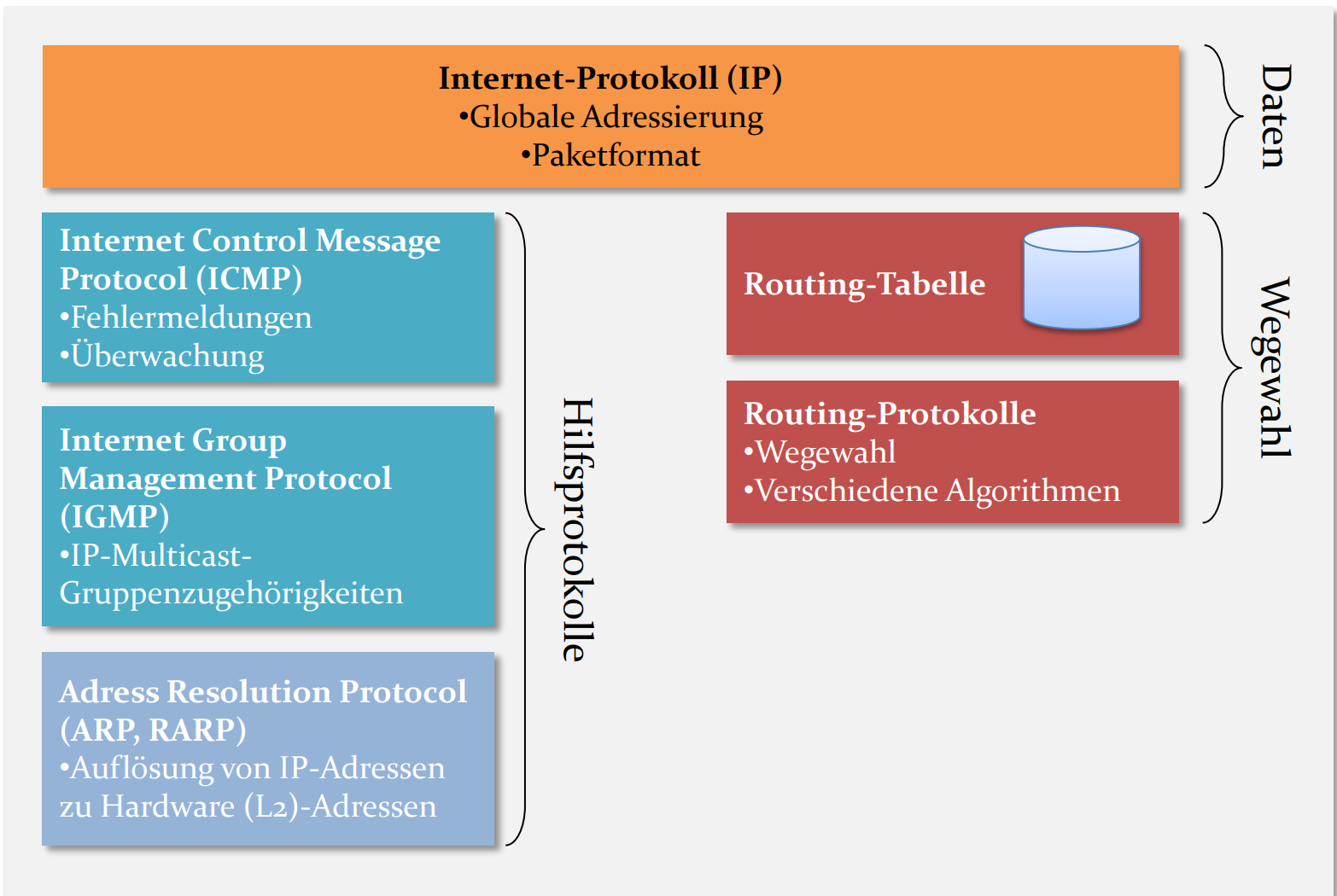
* + - * kleinste Einheit = Nutzer des Internets

Zugangsnetze (AccesNetworks) (WAN)

* + - * Sorgen für Anbindung der Endsysteme, „Anschluss“ an Internet (ADSL-Wählleitung …)

Kernnetze (Internet Backbone) (WAN)

* + - * Transport gewaltiger Datenmenge zwischen einzelnen Routern
      * High-Performance WAN
      * von ISPs (Internet-Service-Provider) und Netzbetreibern
      * ISBs verbinden ihre Netze untereinander und verkaufen einander Transit-Dienste
* Autonome Systeme
  + Gruppe von IP-Netzen und Routern, die von ISP verwaltet werden
  + durch weltweit eindeutige 32 Bit-Nummern (**ASN**) benannt, die von den **RIR** verwaltet werden
  + Routing-Politik
    - Wegewahl wird über **IGP (Interior Gateway Protocol)** vorgenommen
    - Zwischen den AS wird Wegewahl durch Border Router mittels **EGP (External ..)** durchgeführt

**Adressierung und Wegewahl im Internet**

* IP-Adresse für jeden Computer im Internet
  + teilt sich in Netzwerkadresse (Subnetz 🡪 WAN / LAN) und Host-Adresse
* Routing im Internet anhand Netzwerkadresse
  + Ziel Einträge in Routing-Tabelle sind Netzwerkadressen (= Host-Adresse des Routers im eigenen Netz)

**Adressen des Internet-Protokolls**

**IP-Adressen**

* IPv4 4 Oktette (Bytes) bzw. 32-Bit Ganzzahl
  + Aufteilung: Netzwerk-ID (Präfix) und Host-Adresse (Suffix)
  + 9 Adressen / km2 der Erde
  + Darstellung:
    - Jedes Oktett als vorzeichenlose Ganzzahl und durch Punkt getrennt
    - Adressbereich 0.0.0.0 bis 255.255.255.255

IPv4: 194. 28. 225. 10

Binär: 1100010.00011100.11100001.00001010

* IPv6 128 Bit Länge
  + Aufteilung wie IPv4 in Netzwerk-ID und Host-Adresse
  + 667 Billiarden Adressen / mm2 der Erde
  + Darstellung:
    - Je zwei Bytes der Adresse als Hexadezimalwert und durch Doppelpunkt getrennt

IPv6: 2001:0000:02c4:0000:0000:a12b:0001:abc0

Binär: 0000 0100 0000 0001‬: …

* + - Führende Nullen können weggelassen werden:

IPv6 w/o Null: 2001:0:2C4:0:0:A12B:1:ABC0

* + - aufeinanderfolgende Null-Gruppen dürfen 1x/Adresse durch :: abgekürzt werden:

IPv6 gekürzt: 2001:0:2c4::a12b:1:abc0

* + - in URL muss IPv6 in eckige Klammern:

http://**[2001:0:2c4::a12b:1:abc0]**:8080/index.html

* Zusammensetzung:
  + Präfix Netzwork-ID (vordere n Bit der IP) Netzwerkadresse
  + Suffix Interface-ID Rechner/Interface innerhalb des Netzes
* Netzwerkadresse ist weltweit eindeutig, wird koordiniert durch die IANA bzw. RIR
  + Router müssen nur Netzwerkpräfixe wissen, Suffixe werden innerhalb des Netzwerks vergeben

**Subnetze**

* Alle Hosts mit der selben Netzwerkadresse
  + Adresse eines Subnetzes: Präfix mit dem kleinsten Suffix
* Subnetzmaske: Gibt Anzahl n der Präfix-Bits an
  + Anzahl Bit der Netzwerk-ID wird mit /n an Adresse gehängt
  + Präfix besteht bei Subnetzmaske aus 1en, Suffix aus 0en

194. 28. 225. 10/24

IPv4-Adresse: 11000010.00011100.11100001.00001010

Subnetzmaske: 11111111.11111111.11111111.00000000

Präfix|Suffix

* Adressraum: 232-n oder 2128-n mögliche Suffixe
  + zwei Suffixe reserviert:
    - kleinstes Host-Suffix: z.B. xxx.xxx.xxx.0 (bei /24) steht für das gesamte Subnetz (Netzwerkadresse)
      * UND-Verknüpfung aus IP-Adresse und Netzwerkmaske (bitweise)
    - höchstes Host-Suffix: z.B. xxx.xxx.xxx.255 stellt die Brodcast-Adresse dar
* Subnetzmaske mit n bits kann 2n verschiedene Netzwerke bilden (28 = 256,…)
* Aufteilung Netzwerke
  + Netzwerkklassen (**classful addressing**) Klasse aus den erste 4 Bit der IP lesbar
    - Class A /8 = 256 Netze 0000 – 0111
    - Class B /16 Netze 1000 – 1011
    - Class C /24 Netze 1100 – 1101
    - Class D Multicast 1110
    - Class E Reserviert 1111
  + CIDR = Classless Interdomain Routing
    - beliebige Länge der Subnetzmaske (keine Verschwendung von IP-Adressen)
    - Ehemalige Klassen sind Spezialfälle von CIDR
  + in IPv6 keine Klassen mehr

**Vergabe von IP-Adressen**

* Koordination Adressräume
  + Unternehmen / Netzkunde erhält Präfix von ISP
  + ISP erhält Netzwerknummer von zuständiger RIR (Regional Internet Registry) (EU: RIPE NCC)
  + RIR unter Verwaltung von IANA (Internet Assigne Numbers Authority)
* Adressräume können durch verändern der Subnetzmaske (/23 zu /25 = Vervierfachung) aufgeteilt werden

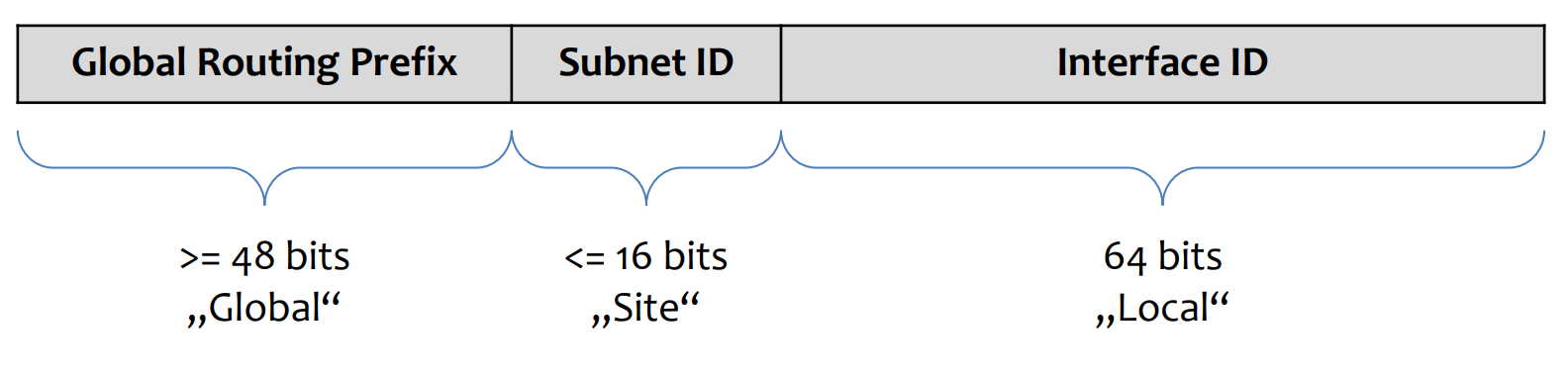
**Private IPv4-Adressen**

* Netzwerknummern, die für private Subnetze ohne Beantragung verwendet werden dürfen (nicht im globalen Internet sichtbar, wegen Eindeutigkeit)
  + Netze ohne Internetanschluss
  + Netze die via NAT ans Internet angeschlossen sind
* Klassen:
  + Class A 10. 0.0.0/8
  + Class B 127. 16.0.0/16
  + Class C 192.168.0.0/16
* NAT (Network Address Translation)
  + private IP Adressen der Hosts eines Subnetzes werden durch NAT in eine oder mehrere global gültige Adressen übersetzt
  + Router versteckt NAT Adressen hinter seiner eigenen, globalen

**Spezielle IPv4-Adressen**

* Addressen über die ganze Netzwerke oder Gruppen von Computern angesprochen werden können
  + Netzwerkadresse Adresse des Netzwerks
    - niedrigste Adresse des Netzwerks (für Routingtabellen)
  + Gerichtete / begrenzte Broadcast-Adresse Adressiert alle Hosts eines Netzwerks/lokalen Netzwerks
    - gerichtet: Suffix sind 1-er Bits (soll alle Hosts im Netzwerk ansprechen)
    - begrenzt: Adresse besteht aus 1-er Bits (255.255.255.255) (z.B. für DHCP-Server)
  + 0.0.0.0-Adresse Unspezifizierte IP-Adresse
    - wird benutzt, wenn Host seine eigene Adresse nicht kennt
  + Loopback-Adresse (localhost) Adressiert eigenen Computer
    - werden dem eigenen IP-Stack zugestellt Adresse: 127.0.0.0/8 localshost: 127.0.0.1
    - für Fehlerdiagnose oder Kommunikation zweier Netzwerkapplikationen auf selbem Computer

**IPv6 Adresstypen:**

* Unspecified (wie in IPv4)
  + ::/128
* Loopback (wie in IPv4)
  + ::1/128
* Scoped
  + Link local
    - wird automatisch für jedes Interface generiert (wird nicht geroutet)
    - fe80:0:0:0:<Interface-Identifier>
  + Unique local unicast
    - private Adressen, wie site local, vgl. private IPv4 Adressen
    - fc00::/7 fd00::/7
  + Global unicast
    - normale, globale IPv6-Adresse
* Multicast

**Routing im Internet**

**Einführung**

* Wegewahl im Internet: Next-Hop Entscheidung
* verschiedene Weiterleitungsverfahren:
  + Datagramm-Routing: Wegewahl für jedes Paket unabhängig vom Ziel (IP)
  + Session-Routing: Wegewahl beim Verbindungsaufbau (ATM oder X.25)
* Komponenten:
  + Routing-Protokoll: Austausch von Routinginformationen zwischen Routern
  + Routing-Algorithmus: Wahl von Wegen im Netz
  + Routing-Tabellen: Halten von Weginformationen (Wegweise für Datenpakete)

**Routingtabelle**

* Aufbau:
  + Next-Hop: Nächster Router auf dem Weg zum Zielnetzwerk (Eingangsadresse)
  + Schnittstelle: Netzwerkinterface des Routers für Next-Hop (Ausgang aus Router)
  + Metrik: Kostenindex aus versch. Faktoren (Hopcount, Kosten, Last, Bandbreite)
    - günstigste Route (Least-Cost) wird gewählt
* Netzwerkfehler: Eintrag in Routingtabelle muss ersetzt werden

**Statisches Routing**

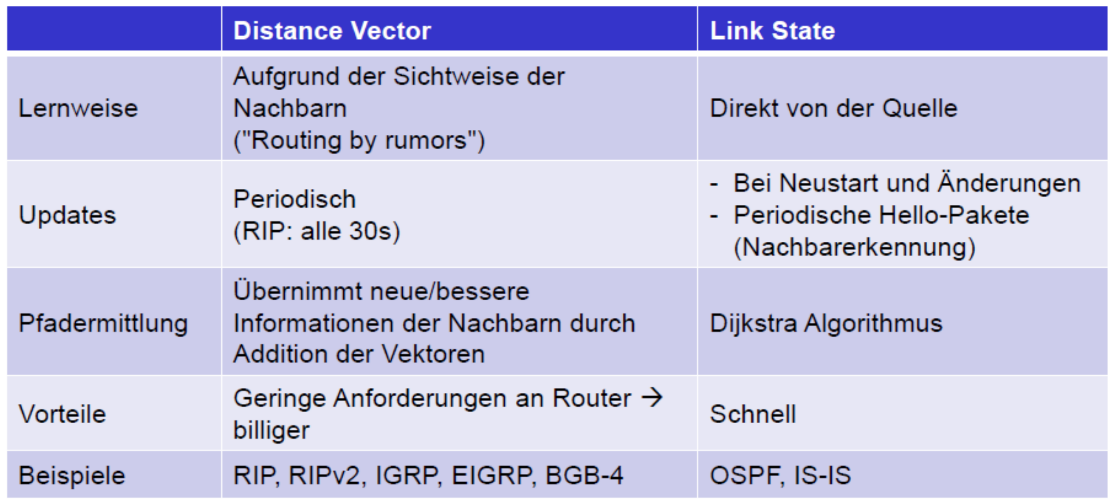
* Erstellung / Pflegen der Einträge in RT von Hand
  + Vorteile: Kein Overhead durch Routing-Protokolle, Kontrolle über einzelne Router
  + Nachteile: Keine Fehlertoleranz, schnell unüberschaubar / komplex

**Distance-Vector-Routing**

* Prinzip: Router speichern Matrix mit bester Metrik, Matrix an andere Router leiten & von anderen lernen
  + Protokolle: RIP, IGRP, BGP-4
  + Nach bestimmter Konvergenzzeit 🡪 Routingtabelle final mit günstigsten Wegen
* Count-to-Infinity Problem: Wegen möglichen Schleifen & Netzwerkfehlern kann Hop-Count unendlich werden
  + Abhilfe:
    - max. Hopcount festlegen, bei „Unendlichem“ Weg wird Router nicht benutzt
    - Split Horizon: Router darf keine der Routen an Interface, über die er gelernt hat, zurücksenden
    - Split Horizon with Poison Reverse: Nachbar nicht erreichbar 🡪 Route wird „unendlich“ gesetzt & vergiftet

**Link-State Routing**

* Prinzip: Alle Router lernen Topologie des Netzes 🡪 eigenen Nachbar lernen und allen anderen mitteilen, Algorithmus zum Berechnen der günstigsten Route
  + Protokolle: OSPF, IS-IS
  + Hello-Pakete an Nachbarn, LSA (Link State Announcements) werden gefloodet
  + kürzeste Wege durch Dijkstra’s Algorithmus
    - eintragen kürzesten Weg von Startknoten in Knoten
    - falls Weg kürzer wird über andere Zwischenknoten 🡪 neuen Weg eintragen & Vorgänger speichern

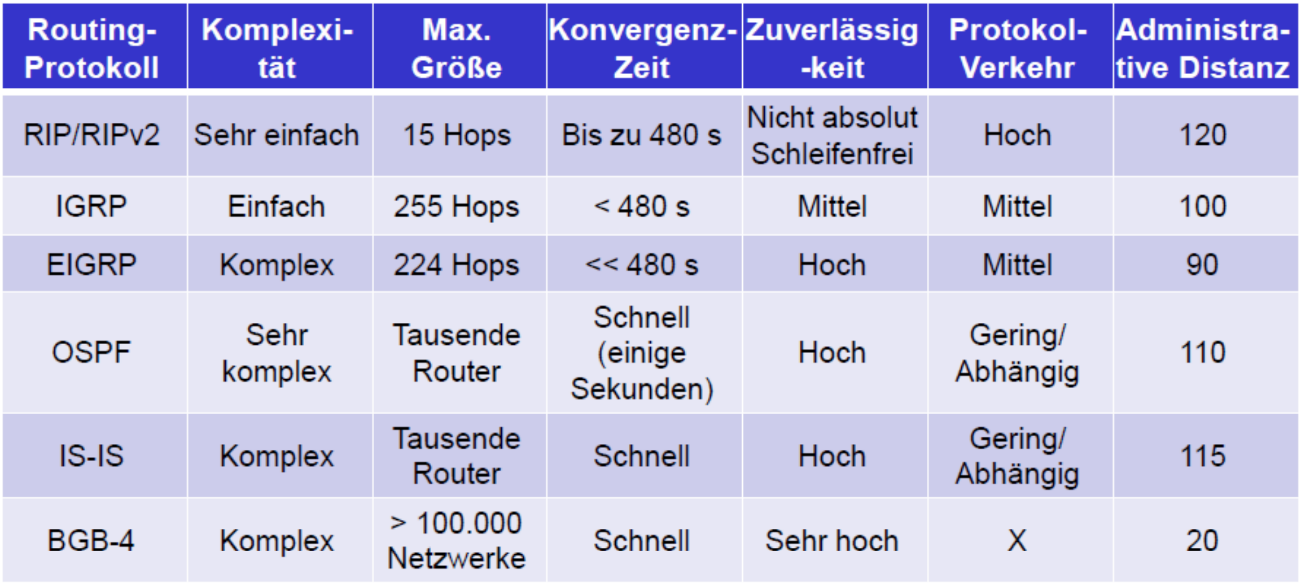




**Autonome Systeme**

* Innerhalb AS: Routing mit IGP (OSPF, IS-IS, RIPv2)
* Zwischen den AS: Routing mit EGP (BGP-4)
  + riesige BGP-RT: Routen oft manuell 🡪 politische / kommerzielle Gründe





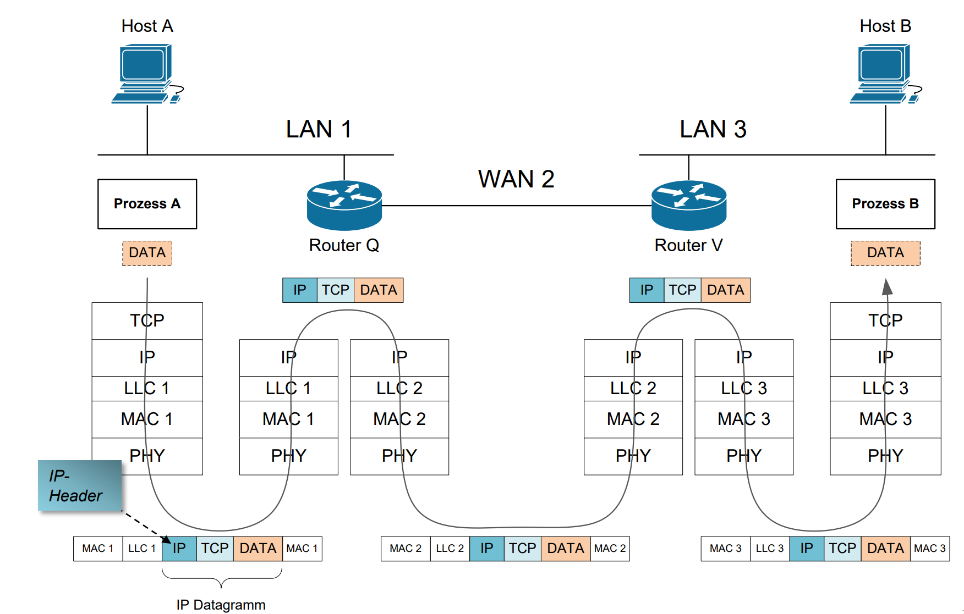
**Das Internet-Protokoll**

**Einführung**

* gemeinsame Sprache aller Internet-Hosts
* besteht aus:
  + Paketkopfaufbau (HeadeR)
  + Adressierung
  + Fragmentierung
  + Verschiedene Dienste
* IP benötigt noch weitere Protokolle:
  + ARP: Adressauflösung
  + ICMP: Kontrollbotschaftern zur Fehlersuche / Netzwerkanalyse

**Eigenschaften des Internet-Protkolls**

* paketvermittelnd, verbindungslos (individuelle Vermittlung von Datagrammen) und ungesichert (Datagramme können verloren gehen, dupliziert werden …)
* Dienstgüte (QoS): Best Effort (Zustellung so gut wie möglich, keine Stau-/Flusskontrolle)



**IP-Paketkopf (Header)**

IPv4:

* Version, Header Length, ToS (Type of Servie), Total Length, Identification, Flags, Fragment Offset, TTL (Time-to-Live), Type, Header Checksum, Source IP Address, Destination IP Address, IP-Options, Padding

IPv6:

* Version, Traffic Class, Flow Label, Payload Length, Next Header, Hop Limit, Source/Destination Address
* Extension Header: Zusätzliche Informationen, nur bei Bedarf angehängt

**Fragmentierung und Reassemblierung**

* Maximum Transmission Unit (MTU) maximale Nutzdatenlänge/Paket innerhalb eines Netzwerks
* Fragmentierung: Aufteilung eines Datagramms in Fragmente < MTU, wird durch Router vorgenommen
* Header-Informationen:
  + Flags: gibt an ob weitere Fragmente folgen oder nicht
  + Offset: Gibt Position im Original-Datagramm an (für Reassemblierung)
* Fragmentierung in Vielfachen von 8 Byte-Größe
* Reassemblierung:
  + Zielhost wartet maximale Zeit lang auf eintreffen fehlender Fragmente
  + bei Eingang des letzten Fragments 🡪 zusammensetzen bzw. reassemblieren
  + wenn Fragmente fehlen werden alle Fragmente des Datagramms verworfen

**IP-Multicast**

* Adressierung einer Gruppe von Teilnehmern
* IP-Adressbereich:
  + IPv4: 224.0.0.0 bis 239.255.255.255
  + IPv6: ff00::/8
* **IGMP (Internet Group Management Protocol)** Verwaltung Gruppenzugehörigkeit

**SourceRouting**

* Vorgabe welche Router ein Datagramm passieren muss
  + Loose: muss unter anderem diese Router passieren
  + Strict: darf keine anderen Router passieren
* Optionsfeld: Loose = Option 3, Strict = Option 9
* Ablauf: 1. Eintrag in Liste ist Zieladresse, beim passieren immer 4 Byte weiterschieben bis Liste leer

**Route Recording:**

* passierte Router werden als Liste im Optionsfeld gespeichert 🡪 zur Rückverfolgung

**Adressauflösung**

* IP-Adresse muss zur Zustellung in MAC-Adresse umgewandelt werden
* Methoden:
  + IPv4: ARP
    - erhält IP, Broadcast (Anfrage Zuteilung IP)
    - Host mit IP teilt MAC-Adresse mit (wenn vorhanden)
    - IP- & MAC-Adresse werden temporär in ARP-Cache gespeichert
  + IPv6: ICMPv6 (bzw. Neighbor Discovery Protocol)
    - Funktionsweise wie ARP, aber verwendet ICMP-Messages
    - bietet zusätzliche Dienste: Router-Discovery, Adressduplikate erkennen, Parametererkennung

**Internet Control Message Protocol ICMP**

* zum Austausch von Fehlermeldungen und Zustandsinformationen
  + Ziel/Port nicht erreichbar
  + Ping (Echo-Anfrage/Antwort)
  + …

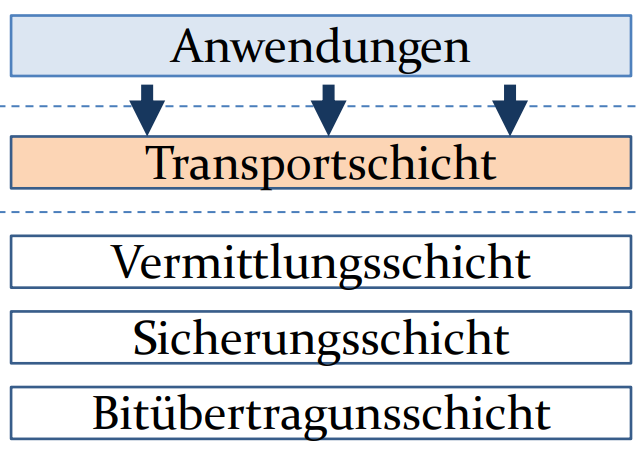
**Transportschicht im Internet**

**Einführung**

* Schicht 4 im OSI-Modell

**Ende-zu-Ende Protokolle**

* Protokoll zwischen Endpunkten einer Kommunikation
* Wegfindung/Kommunikation durch Schicht 3 gewährleistet
  + Transportschichtinstanzen kennen nur Transportschicht der Gegenseite



**Aufgaben der Transportschicht**

* löst anwendungsabhängige Probleme der Netzwerkkommunikation
* Grenze zwischen
  + Benutzern des Netzes (Applikationen)
  + Bereitstellern des Netzes (darunterliegende Schichten)
* Aufgaben
  + Bereitstellung zuverlässigen Dienst zur Übertragung
  + Verbergen der darunterliegenden Komplexität
  + Adressierung von Anwendungen
  + Multiplexing der Daten versch. Anwendungen
  + Bereitstellung Programmier-Schnittstelle (API)

**Anwendungsadressierung**

* Multiplexing
  + Mehrere Anwendungen nutzen gleichzeitig die selbe Netzwerkschnittstelle 🡪 Zuordnung Datenpakete
  + IP kann nicht zwischen verschiedenen Anwendungen unterscheiden (sendet Daten an Computer)
* Transportadressen
  + Anwendungen werden durch Transportadressen adressiert
  + **Transport Service Access Point (TSAP)** 🡪 Ports (Softwareports, keine Hardware Ports)
    - Service Access Point:
      * Zugangspunkt des Dienstes einer Schicht
    - Ports
      * Adressierung der Anwendungen anhand von Portnummern
      * Zieladresse steht im Transportschicht-Header
      * Quellportnummer für Antwort
* Sockets
  + Anwendungsschnittstelle = Socket
  + Socket-Adresse = IP-Adresse + Portnummer

IP:Port 192.168.1.11:1234

* Wahl von Portnummern
  + Festlegung (allgemeine Client-/Server-Architektur)
    - Server wartet auf Port für Anfragen
    - Client-Software verwendet Server-Port für Anfragen
  + Well-known Ports (Standartisierte/Häufig verwendete Anwendungen) (0-1023)
    - Webserver verwenden im Allgemeinen Port 80 = HTTP (TCP)
    - Clients verwenden well-known Ports für Anfragen

**Eigenschaften von Transportprotokollen**

* Verbindungen (TCP)
* Verbindungsloser Programmdienst (UDP/SCTP)
* Zuverlässigkeit (TCP/SCTP) / Fehlererkennung …

**Transmission Control Protocol (TCP)**

* wichtigstes und am häufigsten verwendetes Transportprotokoll im Internet
* Eigenschaften:

Verbindungsorientiert

Zuverlässiger Dienst

Geordnete Datenübertragung (sortierte Übergabe der Daten an Anwendung)

Flusskontrolle (Empfangsfenster)

Stauvermeidung (Sendefenster)

Full-Duplex

Stream-Orientierung (Aufteilen in Segmente / Zusammensetzen übernimmt TCP)

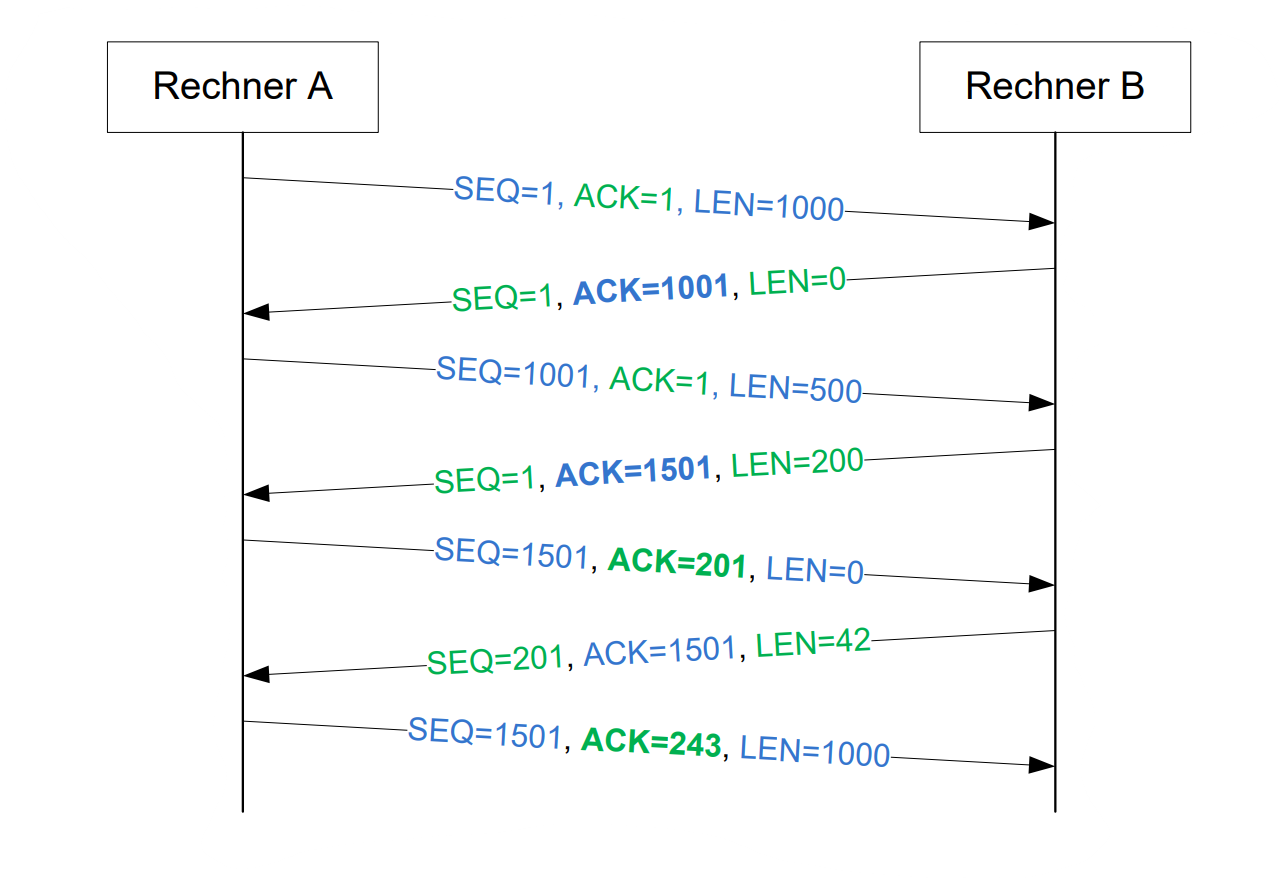
* Paketaufbau:
  + Source/DST Port, Sequence Number (Durchnummerierung nach Bytes), Acknowledgement (Quittierung mit nächster erwarteter Sequenznummer), Header Length, Flags, Window Size (Empfangsfenster), Checksum, Urgent Pointer, Options
* Phasen:

Verbindungsaufbau (beidseitiges Versenden eines Segment’s mit SYN-Flag und ACK)

Datenübertragung (beide Richtungen)

* + - * Versendung mit Seq-Nummer des 1. Bytes, ACK mit nächsten erwarteten Seq-Nummer (Seq + Länge + 1)
      * ACK nur wenn lückenlos übertragen wurde (kein ACK = Retransmit)

Verbindungsabbau (beidseitiges Versenden eines Segment’s mit FIN-Flag und ACK)

* Fast Retransmit
  + ACK wird nach jedem empfangenen Segment gesendet 🡪 bei Lücke wird gleiches ACK 2x gesendet & Retransmit gestartet
* Flusskontrolle / Window Size
  + Puffer für Anwendung hat nur bestimmt große Größe
  + Empfänger teilt Sender mit ACK den verbleibenden Platz im Empfangsfenster mit
  + Sender sendet nur wenn WIN-Größe > 0
* Staukontrolle
  + TCP Sendefenster: nur n unbestätigte Bytes gleichzeitg versenden
  + TCP Slow Start
    - zu Beginn n = 1 MSS (Maximum Segment Size)
    - für jedes empfangene ACK wird n um 1 MSS erhöht
    - Verdopplung von n bis Schwellwert, danach steigt n linear (Congestion Avoidance)
    - Timeout 🡪 n = 1 MSS, CA-Schwelle auf Hälfte der aktuellen Fenstergröße

**User Datagram Protocol (UDP)**

* neben TCP eines der meistverwendeten Transportschichtprotokolle
* Eigenschaften

Verbindungsloser Dienst (vgl. Telegramm)

Unzuverlässiger Dienst (keine Quittierung

Nachrichtenorientiert (jedes Datagramm ist einzelnes IP-Paket)

1-to-many interaction (Nachrichten können an/von einem/mehreren Empfänger/Sender)

* Folgerung
  + einfacher und undkomplizierter als TCP, schnell, hat wenig Overhead
  + unzuverlässige Datenübermittlung (Telefonie, Medienstreaming)
* Headerdaten
  + Source-/DST-Port, Message Length, Checksum (optional)

**Weitere Protokollbeispiele**

* SCTP (stream Control Transmission Protocol)
  + Kompromiss zwischen UDP und TCP, für Reliable Server Pooling
* Scalable TCP
  + wie TCP, aber modifizierter Congestion Control Mechanismus

